Evolution of PCI

A Linux IO Geek's View of HW

Grant Grundler
Hewlett-Packard Company

iod00d (a) hp.com
grundler (a) parisc-linux.org

http://iou.parisc-linux.org/gelato2006/pci_evolution/

Overview

- PCI Configuration Space
- PCI 2.1 64-bit/66Mhz
- PCI 2.2 MSI
- PCI 2.3 MSI-X
- PCI-X 1.0 Split Transactions/133MHz
- PCI-X 2.0 DDR
- PCIe 1.0 Point-to-Point Serial/Full Duplex/AER

- Themes: Resources, Cacheline flows, Bandwidth
- Omitting: PCI Hotplug, Power Management

Interactive talk

(please ask questions)

PCI Local Bus Specification

1992

- ISA Config was not "User Friendly"
- Plug N Pray was not reliable/support nightmare

- Vendor/Device IDs
  - standardized device discovery for drivers

- Base Address Registers
  - I/O Port space is very limited resource
  - MMIO address space standardized non-X86 usage

PCI 2.1 64-bit/66MHz

1995

- 64-bit DMA address
  - x86/PAE could avoid Bounce Buffers
  - Linux had to introduce new DMA api: pci_map_page()
  - RISC arches could bypass IOMMU
    - (Digress: IOMMU out of Style?)
- 64-bit Data Path: More Bandwidth
- 66MHz: quadruple the bandwidth (aka PCI-4X)
  - necessary for GigE, dual port Ultra2 SCSI
- Broken Implementation
  - LSI 53c896 64-bit BAR that wasn't
  - Hidden: many x86 BIOS still only assign 32-bit MMIO

PCI 2.2 Message Signaled Interrupt

1998

- "Inband" message avoids DMA/IRQ races
  - Enables "normal" use to completely avoid MMIO Reads
  - Can substantially improve performance
- Permits multiple vectors directed at one CPU
- Defines 5 bits for vector (32)
- Defined to be an exclusive, unshared interrupt
  - critical to avoiding MMIO reads
- Optional Feature - not widely adopted

PCI 2.3 MSI-X

2002

- Same concept as MSI
- defines 11 bits (2048) for vector
- Permits each vector to target a different CPU
- Requires 64-bit Address (top half can be zero)
- Optional Feature

- Linux support was very x86 centric
  - Processor Interrupt Block hard coded to Oxfee00000
  - Recently fixed in kernel.org by Mark Maule (SGI)
  - SGI needed this for Altix platform
  - "Dead" architectures can too (eg PA-RISC, SPARC, Alpha)

PCI 2.3 MSI-X (cont.)

- Broken implementations made this frustrating
  - e.g. bcm5701, bcm5703/4, Intel 82546GB, et al
  - A few chipsets broken too (e.g. AMD 8131)
  - I'm sure we'll discover more
- Working implementations more common now
  - e.g. Mellanox (IB), Bcm NetExtreme II, Neterion 10GigE
  - Quick survey: about 10 Linux drivers support MSI/MSI-X
  - PCI Vendors finally implementing "Interrupt Steering"

PCI-X 1.0 Split Transactions/133MHz

2000

- Compatible with PCI - same mechanicals
- Requires 64-bit Address/Data support
- Requires MSI or MSI-X (or both)
- Explicit Byte Count for Reads

- Split Transactions allow many requests in flight
  - Similar to SCSI Queue tags
  - Obsoleted "Cacheline Prefetching" schemes in chipsets
    - See "DMA Hints" paper from OLS2003
  - "Deferred Transactions" are a predecessor

- Allowable Disconnect Boundaries (ADBs)
  - Disconnect only allowed on 128 byte ADDRESS boundaries

PCI-X 1.0 (cont.)

- Initiator wait states NOT permitted
  - Initiator must disconnect at ADB
  - Improves bus utilization
- Relaxed Ordering
  - Host driver can enable via PCI-X Command register
  - Device can then specify a mem write be weakly ordered
  - Allows a sort of QoS for DMA flows

PCI-X 2.0 133MHz DDR

2002

- Double Data rate yields 2GB/s BW
  - Solves the immediate bottleneck for 10GigE
- Quad Data Rate also defined in Spec
- cycle-by-cycle ECC (not CRC)
  - single bit errors can be transparently corrected

PCIe 1.0

2003

- NOT HW Compatible with PCI or PCI-X
- Requires MSI or MSI-X (or both)
  - PCI Vendors are getting MSI/MSI-X right!
- Requires 64-bit MMIO support

- Point-to-Point Serial interconnect
  - can have longer traces
  - multiple lanes (more BW!)
  - can build fabrics with switches

- Full Duplex
  - DMA Flows are more efficient
  - Double the BW for many workloads

PCIe 1.0 (cont.)

- Advanced Error Reporting (AER)
  - Standardized PCI device error report
  - SERR signal wasn't sufficient for recovery
  - Linux has limited support for AER (Intel)
- LCRC : Link CRC for link retry/recovery
- ECRC : End-to-End CRC guards data integrity across switches

For more info see...

http://www.pcisig.com/
http://www.ia64-linux.org/
http://www.hp.com/linux

http://iou.parisc-linux.org/ols_2003/